



ATTORNEY DOCKET No. 114596-31-000127BS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Signal No.: 09/672,424 Confirmation No.: 7407
Applicant: Gurjeet S. Saund, et al.
Title: COMPLEX INSTRUCTION SET COMPUTER
Filed: September 28, 2000
Art Unit: 2183
Examiner: E. Coleman
Atty. Docket: 114596-31-000127BS
Customer No. 38492

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF DR. DAVID R. LEVINE

I, David R. Levine, declare as follows.

1. I hold the following degrees:

- Ph.D., Computer Science, Stanford University (1973)
- M.S., Computer Science, Stanford University (1968)
- B.A., Physics, Harvard University (1965)

2. My professional experience includes:

- 38 years working as a computer scientist in a variety of areas, including operating systems, programming language design, compilers and optimization, and instruction set architecture design
- Assistant Professor of Computer Science at Rutgers University
- Assistant Professor of Computer Science at Boston College
- Adjunct Lecturer in Computer Science at the University of Massachusetts
- Co-inventor on U.S. Patent No. 6,332,188 for features in the instruction set architecture of Analog Devices' TigerSHARC processor

- Member of the ANSI Fortran Language Standards committee (X3J3) (representing Hewlett-Packard Company), and also the ANSI Parallel Computing Committee (X3H5), both committees relating to parallel execution
 - At Stanford, I was one of four key designer/implementers of a multi-user timesharing system. One of my specific responsibilities involved the design, implementation and maintenance of interrupt handlers, and their interaction with general process scheduling.
 - I am currently working on an operating system project, with responsibilities in the area of interrupt and exception handlers.
3. In preparing this declaration, I have reviewed the following portions of the prosecution history for application serial no. 09/672,424, and other materials:
- Office Action of 6/19/2006.
 - The specification of this application
 - U.S. Pat. No. 5,729,728, Colwell, method and Apparatus for Predicting, Clearing and Redirecting Unpredicted Changes in Instruction Flow in a Microprocessor
 - U.S. Pat. No. 5,404,473, Papworth, Apparatus and Method for Handling String Operations in a Pipelined Processor
 - The materials attached as exhibits to the paper of March 28, 2006: (a) various web pages discussing "microcode," (b) excerpts from the Intel Architecture Software Developer's Manual, Vol. 2.
 - The online version of MPEP §§ 2111-2111.01, describing the rules of claim interpretation, including the principles of "broadest reasonable interpretation consistent with the specification," "consistent with the interpretation that those skilled in the art would reach," and the "ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention"
4. I also rely on my personal knowledge in the art, and the plain meaning given terms in the art.
5. I note that the Office Action is very difficult to understand, for example, because a number of sentences are grammatically malformed, and much of the technological reasoning is rather removed from the ordinary understanding in the art. I have made a good faith attempt to understand and respond to the Examiner's reasons as stated in the Office Action. I reserve the right to opine further based on any future explanation the Examiner or others may offer.

III. Colwell's Microcode is NOT "Architecturally Exposed"

6. I understand that Examiner Coleman expressed the view that "architecturally exposed" embraced everything in "the system" of Colwell '728 and Papworth '473, including resources only available to microcode. This is not correct.
7. The ordinary understanding in the art is that the "architectural" boundary of a computer divides those machine resources that are available to machine language programs from those resources that are not. The resources that are "architecturally exposed" are generally kept constant across different implementations of the computer. Those which are not exposed, *i.e.*, invisible to machine language programs, may be varied, thus providing implementations of different cost, speed, etc. (There are narrow exceptions to these general characterizations, but they do not come into play here.) Under the ordinary understanding of the relevant terms of art in the late 1990's to 2000 and as would be understood in the context of Colwell '728 and Papworth '473, resources used in microcode that are not exposed to machine language programs are not "architecturally exposed." A "principle of operation" of an "architecture," at least in the context of Colwell '728 and Papworth '473, is to keep a number of things in "the system" not "architecturally exposed."
8. The technological analysis in ¶ 3 of the Office Action is incorrect. Colwell's "uopcode" and "mu," fields and "static" bit are fields of microinstructions or only accessible to microinstructions. These features are not "architecturally exposed" in or to macrocode (also known as "machine language," both of which are equivalent to assembly language programs). Exposing them in or to macrocode would change the principle of operation of the prior art, and would render the prior art less reliable and unsuitable for its intended purpose. My view is reinforced by the Intel Architecture Software Developers' Manual: the "branch" instructions described in the Intel book are not like the "macrobranch" instruction described by Colwell '728, and do not include "uopcode" and "mu" fields.
9. To "allow the user or programmer in the Colwell and Papworth system to directly use microinstruction [sic]" would not be desirable. The modification proposed in the Office Action violates the "principle of operation" of the Colwell '728 and Papworth '473 references.

IV. Colwell's "Macrobranches" are Microcode that is NOT "Architecturally Exposed"

10. The technological analysis in ¶¶ 3-7 of the Office Action is incorrect. In the Colwell '728 patent, a "macrobranch" is a microcode instruction. Colwell's and Papworth's microcode, microbranches, macrobranches, "mu" field, "uopcode" field, and "static" bit are not available to ordinary machine language, architectural-level programmers, and are not "architecturally exposed."

V. "Instruction Boundaries" are NOT "Iteration Boundaries"

11. The explanation of ¶ 15 of the Office Action relating to "turning off" indicators is simply wrong. Colwell's computer relies on these bits having particular values at particular times for correct and efficient execution. The Office Action's proposal to turn certain bits off, for reasons not expressed in Colwell '728, would render Colwell '728 unsuitable for its intended purpose.
12. Colwell '728 explains that he must provide exactly one beginning marker, and exactly one end marker per instruction, because certain actions – such as updating the IP, enabling and disabling interrupts, committing intermediate results to architecturally-visible state, etc. – must occur exactly once per macroinstruction. *E.g.*, Colwell '728, col. 24, line 11, col. 25, line 5, col. 25, lines 35-37.
13. If either Colwell '728 or Papworth '473 were modified by substituting intra-instruction iteration processing for instruction boundary processing, or vice-versa, or if bits were "turned off" as the Office Action seems to suggest in its discussion of claim 13, the computer would execute inefficiently or incorrectly.

VI. Conclusion

14. I express no opinion on any issue not expressly set forth above, and reserve the right to opine further if necessary. The opinions I have stated are specific to the particular facts before me today. In some cases, I have opined to a general rule that might have exceptions, but no exceptions that I know of are relevant to the issues before me today.

15. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Respectfully submitted,

Dated: Oct 13, 2006

By: David R. Levine

Dr. David R. Levine